

Vorwort

The Analytical Engine is therefore a machine of the most general nature.

Charles Babbage (1864)

Dies ist ein einführendes Lehrbuch der Informatik. Es umfaßt den Stoff, der typischerweise im ersten Studienjahr an Universitäten in 1–2 Vorlesungen wie „Informatik I“ und „Informatik II“ gelehrt wird. Es geht zentral um Grundkonzepte von (objektorientierten) Programmiersprachen und von Algorithmen, und zur ihrer Umsetzung wird durchgehend Modellierung mit der Sprache UML und Programmierung mit der Sprache Java eingesetzt. Dieses Buch ist also weder ein Java-Handbuch noch ein Schnellkurs im Programmieren. Der zentrale Stoff wird ergänzt durch eine Übersicht über die Architektur eines Computers am Beispiel eines modernen PC-Systems und eine Einführung in die mathematisch-logischen Grundlagen der Informatik. Unter der URL

www-sr.informatik.uni-tuebingen.de/InfoBuch

ist eine Web-Seite zu diesem Buch eingerichtet. Dort sind u. a. Foliensätze für Dozenten zu finden.

Heute ist der Einsatz von Computern nicht mehr auf das klassische Gebiet des technisch-wissenschaftlichen Rechnens konzentriert, sondern dringt auf breiter Front in alle Bereiche von Wissenschaft, Wirtschaft und Gesellschaft vor. Die neuen Einsatzgebiete, wie z. B. moderne Client-Server-Informationssysteme, verlangen in der Praxis neue Werkzeuge und Methoden, auch wenn die alten rein theoretisch noch genügen würden.

Objektorientierte Software-Entwicklung mit der Modellierungssprache UML, der Programmiersprache Java und neuen integrierten Programmierwerkzeugen wie Eclipse sind die wichtigsten neuen Hilfsmittel, mit denen man der Herausforderung immer vielseitigerer, vielschichtigerer und vernetzter Software begegnet. Objektorientierte Methoden haben in der Praxis wesentlich dazu beigetragen, den Komplexitätsschub in Entwurf, Programmierung und Wartung moderner Systeme in den Griff zu bekommen. Java, als Programmiersprache des Internet bekannt geworden, ist eine moderne objektorientierte Sprache, die sowohl durch klare theoretische Konzepte besticht als auch in der breiten Praxis – von Mobiltelefonen bis zu Großrechnern – Anwendung findet.

Der zentrale Beweggrund für dieses Lehrbuch der Informatik war das Erreichen einer Balance zwischen Theorie und Praxis, also zwischen theoretisch-konzeptuellen Grundlagen und unmittelbar praxisrelevantem Wissen. Dieses Lehr-

buch soll die traditionellen Konzepte, die in der Einführungsvorlesung Informatik im ersten Studienjahr gelehrt werden, um den Gesichtspunkt der Objektorientierung ergänzen und aus dieser Sicht neu präsentieren sowie anhand von ausgewählten Teilen von UML und Java einüben.

Der Leser soll insbesondere

- grundlegende Konzepte der objektorientierten Software-Entwicklung und des Programmierens verstehen,
- mit Java eine moderne objektorientierte Sprache erlernen, die diese Konzepte umsetzt und die auch in der breiten Praxis von Wissenschaft und Wirtschaft in allen Anwendungsgebieten und bei großen komplexen Aufgaben verwendet wird,
- höhere Datenstrukturen, Algorithmen und deren zugrundeliegende Entwurfsmuster anhand des Standardrepertoires der Informatik kennenlernen und
- ein zukunftsfestes Wissen der theoretischen Grundlagen der praktischen Informatik erwerben, um eine Basis für lebenslanges Lernen zu erhalten.

Diese Neuauflage schließt sich an die Vorlesungen Informatik I in Bonn im WS 2002/2003 und Informatik I und II in Tübingen im akademischen Jahr 2003/2004 an. In Tübingen haben wir zum ersten Mal von Anfang an die neue integrierte Entwicklungsumgebung Eclipse eingesetzt.¹ Eine für Anfänger nützliche Einführung in Eclipse hätte den Umfang dieses Buchs gesprengt, aber einige Hinweise konnten in den Text aufgenommen werden.

Neben zahlreichen weiteren Detailverbesserungen verdienen die folgenden Punkte besondere Erwähnung:

- Die Reihenfolge der Kapitel in Teil I wurde verändert und ist jetzt: 2. Rechnerarchitektur – 3. Algorithmen – 4. Datenstrukturen – 5. Objektorientierung. Die neue Anordnung folgt einem *bottom-up* Prinzip und läßt sich gerade zu Beginn schlüssiger lehren.
- Kapitel 2 (Rechnerarchitektur) wurde überarbeitet. Die Behandlung von Zahldarstellungen und Konversionsmethoden wurde ergänzt, insbesondere auch im Teil zu Gleitkommazahlen und dient jetzt auch als natürliche Motivation für Algorithmen im nachfolgenden Kapitel 3.
- Die Flußdiagramme in Kapitel 3 (Algorithmen) wurden auf die Notation von UML Aktivitätsdiagrammen umgestellt. Da UML auch von Eclipse unterstützt wird, lassen sich jeweils das Diagramm und der Java-Code nebeneinander in Eclipse betrachten.
- Viele der UML Klassendiagramme wurden neu gezeichnet und dabei einige Abweichungen vom UML Standard bereinigt.
- Es wurden zahlreiche weitere in den oben genannten Veranstaltungen erprobte Übungsaufgaben aufgenommen.

¹ Dieses Konzept mit dem Namen FOOD (*foundations of object oriented development*) wurde 2003 mit einem *IBM Eclipse Innovation Award* ausgezeichnet (siehe www-sr.informatik.uni-tuebingen.de/food).

Danksagung. Wir danken wiederum allen unseren Mitarbeitern, insbesondere denen, die unsere Vorlesungen betreut haben. In Tübingen sind dies Dr. W. Blochinger, W. Westje und Dr. V. Simonis; Frau E.-M. Dieringer hat wiederum zahlreiche neue Abbildungen angefertigt. In Bonn möchten wir insbesondere G. Sobottka, M. Guthe, D. Goldbach und Frau M. Gnasa für Korrektur- und Ergänzungsvorschläge danken.

Besonders zu Dank verpflichtet sind wir unseren Lesern und denjenigen Kollegen an zahlreichen Hochschulen, die die bisherigen Auflagen für Ihre Vorlesungen verwendet haben; sie haben die neue Auflage erst möglich gemacht. Wertvolle Hinweise verdanken wir Herrn Prof. Dr. B. Eberhardt, Herrn Prof. Dr. R. Klein und insbesondere Herrn Prof. Dr. D. Saupe, der uns auch einige Foliensätze überlassen hat. Wir freuen uns über jede weitere Anregung und sind stets offen für Verbesserungen.

Tübingen, Bonn, im August 2004

W. Küchlin, A. Weber

Aus dem Vorwort zur zweiten Auflage

Bei der ersten Auflage lag unser Hauptaugenmerk auf der Weiterentwicklung des klassischen Stoffs in die Tiefe, zur Objekttechnik hin. Nun haben wir das Buch in die Breite ergänzt, insbesondere in Hinblick auf einen Leserkreis, der an einem Einblick in die Struktur eines Rechnersystems und einer Gesamtsicht auf die Software-Entwicklung, auch in ihren klassischen Teilen, interessiert ist.

Danksagung. Wir danken allen unseren Mitarbeitern, die am Zustandekommen dieser Neuauflage beteiligt waren oder unter ihr zu leiden hatten: In Tübingen sind dies Dr. W. Blochinger, C. Sinz, M. Friedrich, R. Schimkat, G. Nusser und A. Kaiser; Frau E.-M. Dieringer hat einige neue Abbildungen angefertigt. In Bonn möchten wir Herrn G. Sobottka und Frau S. Schäfer für Korrekturvorschläge danken.

Besonders zu Dank verpflichtet sind wir unseren Lesern und denjenigen Kollegen, die die erste Auflage für Ihre Vorlesungen verwendet haben; wertvolle Hinweise verdanken wir Herrn Prof. Dr. B. Eberhardt und Herrn Prof. Dr. L. Voelkel.

Tübingen, Bonn, im Juli 2002

W. Küchlin, A. Weber

Aus dem Vorwort zur ersten Auflage

Dieses Buch fußt auf unseren Vorlesungen Informatik I und II an der Universität Tübingen. Wir schulden allen Dank, die am Zustandekommen dieser Vorlesungen in irgendeiner Form mitgewirkt haben, insbesondere natürlich den Mitarbeitern am Arbeitsbereich Symbolisches Rechnen.

Dr. W. Blochinger hat drei Semester lang verantwortlich die Übungen zu den Vorlesungen organisiert; ein Teil seiner Aufgaben ist auch in dieses Buch eingeflossen. Beiträge für die Übungen kamen auch von Dr. J. Hahn, Dr. B. Amrhein und

S. Müller, sowie von studentischen Tutoren, insbesondere von Ch. Ludwig. Kapitel 9 beruht auf Vorlagen von Dr. J. Hahn.

Für viele Korrekturen und nützliche Anregungen möchten wir Herrn Prof. Dr. U. Güntzer und Herrn Prof. Dr. M. Kaufmann herzlich danken, die eine Vorversion des Manuskripts durchgesehen haben. Teile des Manuskripts wurden ferner von Dr. D. Bühler, Dr. G. Hagel, A. Kaiser, Dr. Th. Lumpp, P. Maier, G. Nusser, R. Schimkat, C. Sinz und Dr. A. Speck korrekturgelesen.

Tübingen, Darmstadt, August 2000

W. Küchlin, A. Weber

These discussions were of great value to me in several ways. I was thus obliged to put into language the various views I had taken, and I observed the effect of my explanations on different minds. My own ideas became clearer, and I profited by many of the remarks made by my highly-gifted friends.

Charles Babbage (1864)